

# Konzeptuelle Modellierung partieller SOA

Andree Teusch  
Elmar J. Sinz

Veröffentlicht in:  
Multikonferenz Wirtschaftsinformatik 2012  
Tagungsband der MKWI 2012  
Hrsg.: Dirk Christian Mattfeld; Susanne Robra-Bissantz



Braunschweig: Institut für Wirtschaftsinformatik, 2012

# Konzeptuelle Modellierung partieller SOA

**Andree Teusch**

Universität Bamberg, Lehrstuhl für Wirtschaftsinformatik,  
insb. Systementwicklung und Datenbankanwendung, 96052 Bamberg,  
E-Mail: andree.teusch@uni-bamberg.de

**Elmar J. Sinz**

Universität Bamberg, Lehrstuhl für Wirtschaftsinformatik,  
insb. Systementwicklung und Datenbankanwendung, 96052 Bamberg,  
E-Mail: elmar.sinz@uni-bamberg.de

## Abstract

In diesem Beitrag wird ein Ansatz zur konzeptuellen Modellierung partieller SOA anhand des Aufgabenmodells begründet. Das Aufgabenmodell zeigt, wie Choreografie / Orchestrierung aus der Außensicht / Innensicht von Aufgaben ableitbar sind, wie sich Arten von Services anhand der Trennung zwischen Aufgabenobjekt und Lösungsverfahren unterscheiden lassen und wie sich generell Schnittstellen zwischen partieller SOA und dem Rest der Anwendungssysteme systematisch ableiten lassen. Die Entwicklung partieller SOA wird damit zu einer mehrere Ebenen umfassenden konzeptuellen Modellierung. Die Anwendung verdeutlicht eine Fallstudie, bevor vor dem Hintergrund des Standes der Literatur eine kurze Diskussion des Modellierungsansatzes stattfindet.

## 1 Einführung

Sofern in einem Unternehmen nur bestimmte Bereiche gemäß dem Prinzip der service-orientierten Architektur (SOA) realisiert werden, während andere Bereiche durch herkömmliche Anwendungssysteme (AwS) unterstützt werden, liegt eine partielle SOA vor. Primäres Ziel für eine partielle SOA ist zumeist, das Potential für eine Veränderung im Inneren des Systems oder in seiner Umwelt durch zielorientierte Anpassung seines Verhaltens oder seiner Struktur zu erhöhen, kurz die Flexibilität [2] zu steigern.

Ein Beispiel aus der Domäne Projektstellenbesetzung ist, dass die Bereiche *Personalbesetzung* und *Mitarbeiterqualifizierung* auf Aufgabenebene z. T. oft die Leistungen wechselnder externer Dienstleister einbeziehen und somit auf Aufgabenträgerebene aufgrund ständig wechselnder Anforderungen und Lösungsalternativen hohen Flexibilitätserfordernissen ausgesetzt sind. Die Bereiche *Personal-* und Fortbildungswesen hingegen

werden aufgrund deutlich geringerer Flexibilitätsbedarfe durch tradierte Human-Resource- und Learning-Managementsysteme betrieben.

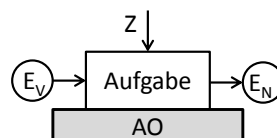
Die Abgrenzung von SOA an betrieblichen Objekten und die modellgetriebene Entwicklung von SOA rücken das Thema in den Bereich der konzeptuellen Modellierung. Der Ansatz zum fachlichen Entwurf partieller SOA auf der Grundlage von Geschäftsprozessmodellen wurde bereits in [17] veröffentlicht. Im Folgenden steht die Begründung des Modellierungsansatzes im Vordergrund. Hierbei werden erläutert und veranschaulicht, wie anhand des Modells der *Aufgabe* sowohl die Spezifikation von Geschäftsprozessmodellen, als auch die Spezifikation der diese Geschäftsprozessmodelle unterstützenden und partiell serviceorientierten Anwendungssysteme hergeleitet werden können.

Zum Aufbau des Papiers: Der Modellierungsansatz wird im nächsten Abschnitt 2 ausgehend vom Modell der Aufgabe aus Außensicht vorgestellt. Dadurch wird der Bereich einer partiellen SOA abgesteckt. Durch den Perspektivenwechsel zur Innensicht des Modells der Aufgabe wird dann auf die spezifische SOA-Realisierung Bezug genommen und diese wird modellgetrieben abgeleitet. Abschnitt 3 wendet den Modellierungsansatz auf ein Beispiel aus der Personalwirtschaft an. Der Stand der Literatur wird in Abschnitt 4 betrachtet, um dann in Abschnitt 5 Vor- und Nachteile, Erweiterungen und Restriktionen des Modellierungsansatzes diskutieren zu können.

## 2 Vom Aufgabenmodell zur SOA

Der Begriff *Aufgabe* ist einer der grundlegendsten der Wirtschaftsinformatik. Alles zweckbezogene menschliche Handeln findet in Aufgaben statt ([16], S. 43). Das betriebliche Informationssystem mit seinen wesentlichen abzustimmenden Komponenten wurde als Mensch-Aufgabe-Technik-System erkannt ([13], S. 173). Durch Gleichsetzung von Mensch und Technik im Rang wurde daraus das betriebliche Informationssystem mit Aufgabenebene (Aufgabe) und Aufgabenträgerebene (Mensch und Maschine) ([10], S. 2 ff.).

Aus Außensicht betrachtet, lässt sich eine Aufgabe durch ein Aufgabenobjekt (AO), ein Ziel (Z) sowie durch Vor- ( $E_V$ ) und Nachereignisse ( $E_N$ ) beschreiben. Letztere lösen eine Aufgabendurchführung aus bzw. resultieren aus der Aufgabendurchführung (Bild 1) ([10], S. 96). Z geben gewünschte Ausprägungen von AO an. Ein Beispiel aus der Personalwirtschaft ist: AO ist die Menge der vorhandenen Qualifikationen, Z die Ausrichtung der Qualifikationen an zukünftigen Projekten.

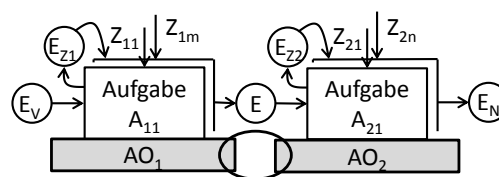


**Bild 1:** Außensicht auf eine betriebliche Aufgabe

Die explizite Berücksichtigung des AO einer Aufgabe macht das Aufgabenmodell zu einer geeigneten Basis zur Ableitung von SOA und führt zu einer Abgrenzung gegenüber anderen SOA-Ansätzen, bei denen die den Services zugrundeliegenden Datenobjekte und ihre Beziehungen i. d. R. unberücksichtigt bleiben. Weshalb die Betrachtung von AO bei der fachlichen Ableitung von SOA relevant ist und wie SOA auf Basis des Aufgabenmodells

abgeleitet werden können lässt sich anhand des Zusammenspiels betrieblicher Aufgaben erklären: Normalerweise ist eine initiale Aufgabe zu groß, um das betriebliche Geschehen hinreichend detailliert beschreiben zu können. Aus diesem Grund wird sie zerlegt. Hierfür bestehen zwei Möglichkeiten: Bei einer Untergliederung nach dem Verrichtungsprinzip werden verwandte Arbeitsabläufe an unterschiedlichen Objekten zusammengefasst, während bei einer Untergliederung nach dem Objektprinzip betriebliche Objekte abgegrenzt werden, die dann um die zugehörigen Arbeitsabläufe ergänzt werden ([10], S. 72 f.).

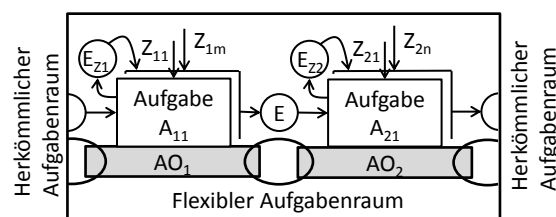
Mit Blick auf die Dominanz der betrieblichen Objekte im Vergleich zu den eher volatilen Verrichtungen wird das Objektprinzip gewählt. Das Aufgabenobjekt AO wird in die beiden Aufgabenobjekte  $AO_1$  und  $AO_2$  zerlegt. Anschließend wird  $Z$  in die Zielgruppen  $\{Z_{11}, \dots, Z_{1m}\}$  und  $\{Z_{21}, \dots, Z_{2n}\}$  und korrespondierend in die Aufgaben  $\{A_{11}, \dots, A_{1m}\}$  und  $\{A_{21}, \dots, A_{2n}\}$  zerlegt (Bild 2).



**Bild 2: Aufgabenzerlegung nach dem Objektprinzip**

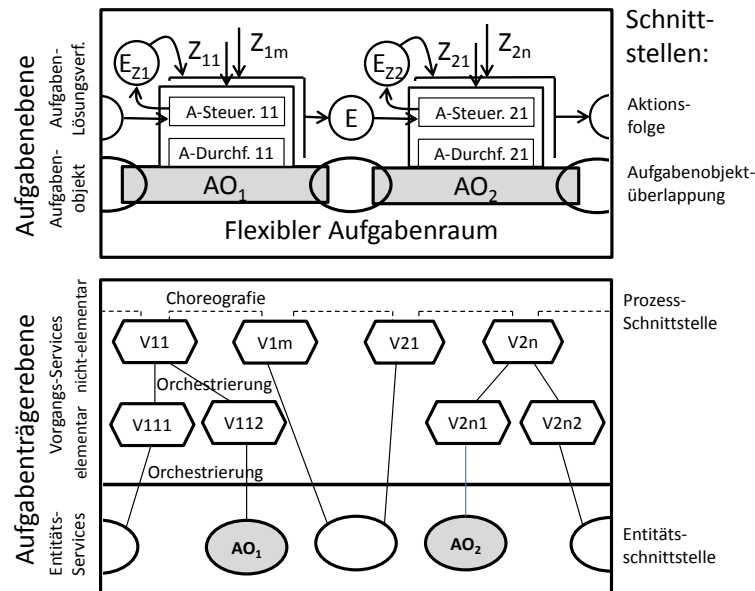
Zwei Aufgaben, die hintereinander ablaufen, sind entweder auf dem gleichen Aufgabenobjekt definiert und tauschen darüber ihre Argumente aus (Aufgaben  $A_{11}$  und  $A_{1m}$  auf Aufgabenobjekt  $AO_1$ , Ereignis  $E_{z1}$ ) oder auf unterschiedlichen Aufgabenobjekten und tauschen über die Überlappung der Aufgabenobjekte ihre Argumente aus (Aufgabe  $A_{1m}$  auf Aufgabenobjekt  $AO_1$ ,  $A_{21}$  auf  $AO_2$ , Ereignis  $E$ ). Damit werden gleichzeitig mit der Zerlegung von AO Schnittstellen zwischen  $AO_1$  und  $AO_2$  sichtbar ([10], S. 632f). Es gilt generell  $AO = AO_1 \cup AO_2$  und  $AO_1 \cap AO_2 \neq \emptyset$ . Das bedeutet, dass bestimmte Attribute in abgestimmter Form in beiden Teilaufgabenobjekten vorkommen werden. Überlappungen bei den Zerlegungsprodukten von  $Z$  werden ebenfalls in Kauf genommen. Sie führen zu Überlappungen bei den Lösungsverfahren und sind ein Preis für die nachrangige Zerlegung von Zielen. Das Nach- / Vorereignis  $E$  markiert den Übergang zwischen den beiden Aufgabendurchführungen.

Da nicht für das gesamte Aufgabensystem ein hoher Flexibilitätsbedarf unterstellt werden soll, ergeben sich ein *flexibler Aufgabenraum*, ein *herkömmlicher Aufgabenraum* und Schnittstellen dazwischen (Bild 3). Diese Abgrenzung kann erst ab einer Zerlegungsstufe erfolgen, ab der jedes Zerlegungsprodukt in genau einem der beiden Aufgabenräume enthalten ist. Bezüglich der Schnittstellen kann nur über die im flexiblen Aufgabenraum enthaltenen Teile etwas ausgesagt werden, es gilt  $AO_1 \cup AO_2 \subseteq AO$ .



**Bild 3: Abgrenzung der Aufgabenräume**

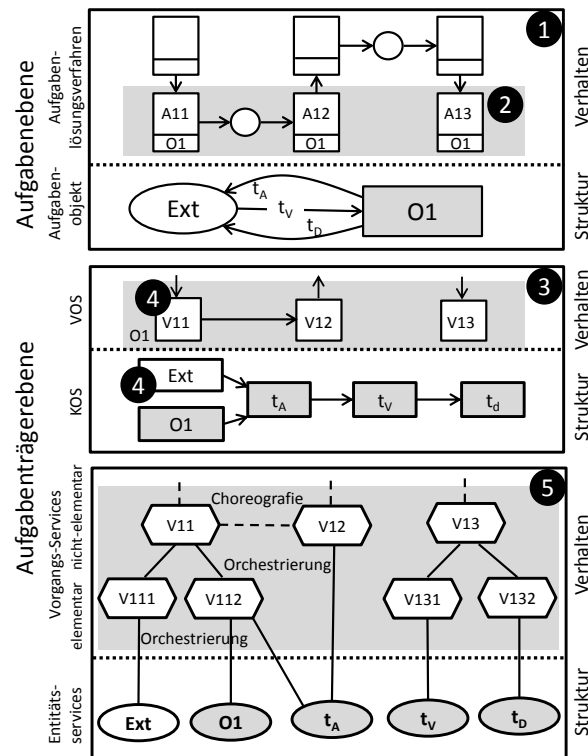
An dieser Stelle kann nun auf die Realisierung der Aufgaben übergegangen werden. Dies erfordert eine Betrachtung der Innensicht der Aufgabenebene (Bild 4) ([10], S. 97). Die Lösungsverfahren der Aufgaben enthalten eine Aktionendurchführung und eine Aktionensteuerung. Zusätzlich wird eine Aufgabenträgerebene angefügt, die auf die gewählte Form der service-orientierten Architektur Bezug nimmt. Die Aufgabenträgerebene stellt die Lösungsverfahren für die Aufgaben zur Verfügung.



**Bild 4: Aufgaben- und Aufgabenträgerebene**

Geeignete Prinzipien zur Koordination der Lösungsverfahren sind die hierarchische und nicht-hierarchische Koordination ([10], S. 67 ff.). Diese Prinzipien finden sich auch in einer SOA wieder: Der hierarchisch koordinierte Kontrollfluss entspricht hier einer Orchestrierung, während der dezentrale Nachrichtenaustausch beteiligter Objekte nicht-hierarchisch in Form einer Choreografie durchgeführt wird [20]. Da die Aufgaben  $A_{1m}$  und  $A_{21}$  aus Außensicht autonom sind, kommt für die Koordination der Lösungsverfahren  $V_{1m}$  und  $V_{21}$  nur die *Choreografie* in Frage. Etwas anderes ist die Koordination der Teil-Lösungsverfahren, die auch auf dem Wege der *Orchestrierung* erfolgen kann. Der Zugriff auf ein Aufgabenobjekt erfolgt durch die Aktionen, welche in Form von entsprechenden Schnittstellen bereitgestellt werden.

An dieser Stelle soll der Ansatz für umfangreichere Schemata handhabbar gemacht werden (Bild 5). Hierzu dient der SOM-Ansatz (Semantisches Objektmodell) ([10], S. 192 ff.), der auf der Aufgabenmetapher beruht und die Aufgabenobjekte in Form eines Interaktionsschemas (IAS), die Lösungsverfahren der Aufgaben (Aktionensteuerung und -durchführung) durch ein Vorgangs-Ereignis-Schema (VES) beschreibt. Die Ableitung der Serviceidentifikation erfolgt jedoch indirekt. Aus den beiden Darstellungen lassen sich ein Konzeptuelles Objektschema (KOS) und ein Vorgangs-Objektschemas (VOS) ableiten, die dann die Grundlage für die Serviceidentifikation bilden.

**Bild 5: Modellgetriebene Zerlegung**

Damit lassen sich die in [17] eingeführten und in Tabelle 1 gezeigten fünf Schritte des Modellierungsansatzes greifbar darstellen:

Schritt	Aktivität	Bemerkung
1	SOM-GP-Modell	Modellierung IAS / VES. Die Ermittlung der Flexibilitätsbedarfe für die einzelnen Bereiche des GPs geht dem Ansatz voraus.
2	Abgrenzung des SOM-GP-Modells / Kartierung der umgebenden AwS	Mapping der im GP-Modell abgrenzbaren flexiblen und herkömmlichen Aufgabenräume auf einzelne AwS. Alle Aufgaben eines betrieblichen Objekts werden durch genau ein AwS automatisiert, ein AwS kann umgekehrt mehrere betriebliche Objekte unterstützen. Hierdurch lassen sich die fachlichen Beziehungen zwischen AwSen vollständig aus dem GP-Modell heraus erklären.
3	Fachliche Spezifikation der SOA	Spezifikation VOS / KOS für die SOA. Grundlage für die modellgetriebene Ableitung der Ergebnisse von Schritt 4 und 5.
4	Externe Schnittstellen des SOA-AwS	Spezifikation der erforderlichen externen Prozessschnittstellen in Form von Vorgangs-Services (Basis der Ableitung: VOS). Identifikation und Ableitung von aufgrund überlappender Aufgabenobjekte benötigter Datenschnittstellen in Form von Entitäts-Services (Basis der Ableitung: KOS).
5	Serviceidentifizierung	Spezifikation von Entitäts-Services anhand der im KOS repräsentierten Aufgabenobjekte. Identifizierung elementarer Vorgangs-Services anhand der im VOS berücksichtigten einzelnen betrieblichen Aufgaben. Ableitung nicht-elementarer Vorgangs-Services anhand der Vorgangsnetze betrieblicher Objekte.

**Tabelle 1: Fünf Schritte zum Entwurf partieller SOA**

### 3 Fallstudie

Die Anwendung der fünf Schritte zur modellbasierten Ableitung einer partiellen SOA, ausgehend von einem SOM-GP-Modell, soll im Folgenden anhand eines Fallbeispiels aus der Domäne *Personalwirtschaft* demonstriert werden. Ausgangspunkt hierfür ist die in Bild 6 dargestellte unternehmerische Gesamtaufgabe der Projektstellenbesetzung.

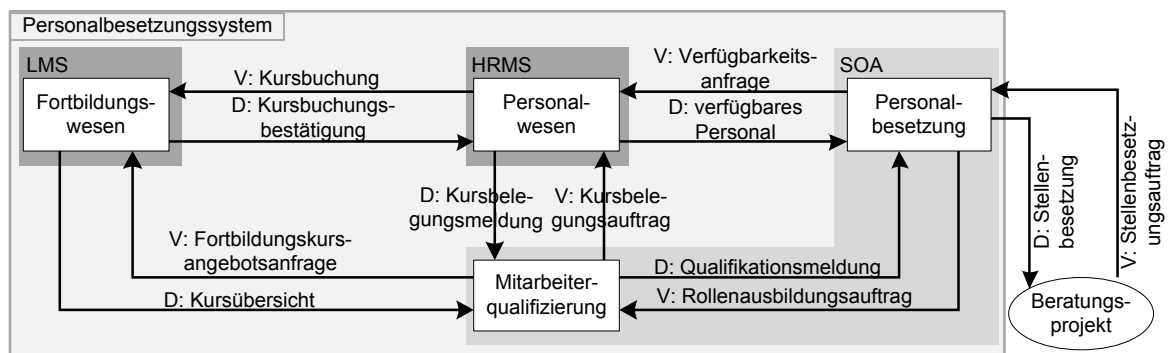


**Bild 6:** Unternehmerische Gesamtaufgabe in Form eines IAS

Leistungsempfänger des Personalbesetzungssystems eines großen Beratungsunternehmens sind spezifische Kundenprojekte des Unternehmens. Die der Gesamtaufgabe zugrundeliegende Leistung besteht in der Besetzung offener Projektrollen auf den Beratungsprojekten mit intern verfügbaren Mitarbeitern (MA). Sofern hierbei ein zur Verfügung stehender MA dem Anforderungsprofil der zu besetzenden Rolle genügt, wird dieser direkt dem anfragenden Projekt zugewiesen. Steht hingegen kein MA zur Verfügung, der das geforderte Anforderungsprofil erfüllt, so wird zunächst ein MA per interner Weiterbildung für die ausgeschriebene Rolle qualifiziert, bevor er dem Projekt zugewiesen wird.

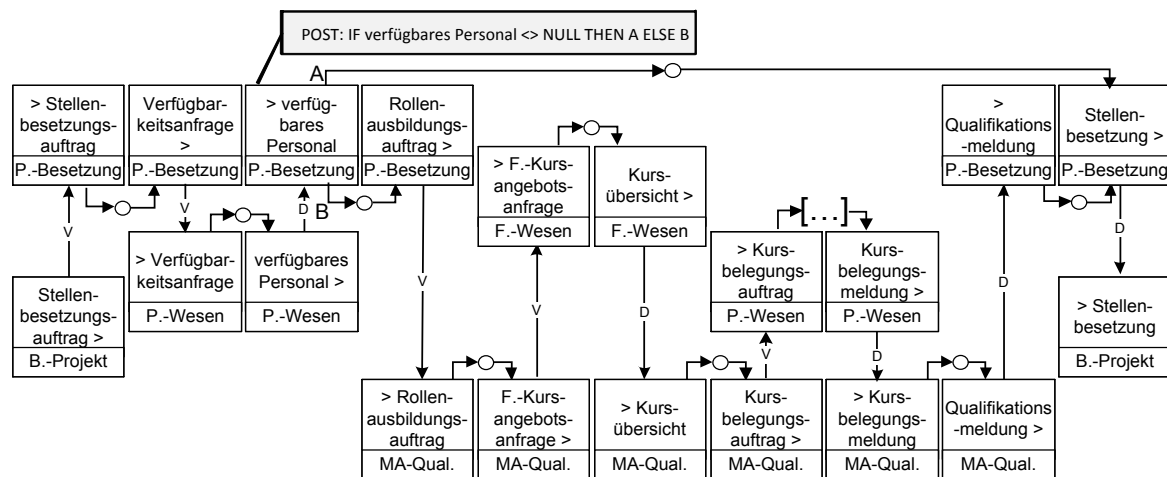
Die Ableitung einer partiellen SOA für das geschilderte Szenario erfolgt anhand der folgenden fünf Schritte:

- (1) Bild 7 stellt die Struktursicht auf den genannten Geschäftsprozess in Form eines IAS dar. Bild 8 zeigt die zugehörige Ablaufsicht in Form eines VES. Beide Diagramme zusammen entsprechen einem SOM-GP-Modell auf detaillierter Zerlegungsstufe.



**Bild 7:** Detaillierte Struktursicht des GPs in Form eines IAS

Das IAS in Bild 7 umfasst die objektorientierte Zerlegung der Gesamtaufgabe *Personalbesetzungssystem* in die betrieblichen Objekte *Personalbesetzung*, *Personalwesen*, *Mitarbeiterqualifizierung* und *Fortbildungswesen*. In Bezug auf das AO findet somit eine Zerlegung in vier Teil-AO statt, die Schnittstellen zwischen diesen Teil-AO werden durch betriebliche Transaktionen zwischen den betrieblichen Objekten realisiert. Zur Erbringung der Gesamtaufgabe müssen die betrieblichen Teilaufgaben in Form von Vorgängen ausgeführt und über einen ereignisgesteuerten Ablauf koordiniert werden. Bild 8 zeigt diese Ablaufsicht in Form eines VES.

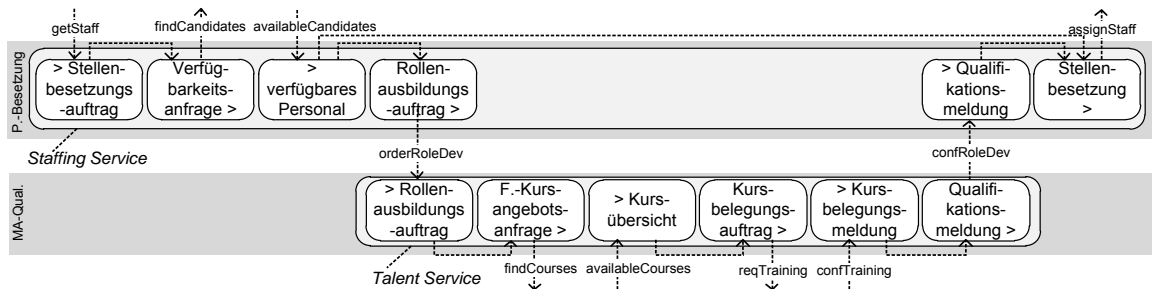


**Bild 8:** Detaillierte Ablaufsicht des GPs in Form eines VES

Zum Beispiel ist die Aufgabe *>verfügbares Personal* mit einer Post-Condition versehen, aufgrund der es entweder bei der Aufgabe *Stellenbesetzung>* oder bei der Aufgabe *Rollenbesetzungsauftrag>* weiter geht.

- (2) Zur modellgetriebenen Ableitung einer partiellen SOA ausgehend von einem GP-Modell werden zunächst die Systemgrenzen der SOA festgelegt. Aus Aufgabensicht ist somit der *flexible Aufgabenraum* von dem *herkömmlichen Aufgabenraum* abzugrenzen. Grundlage dieser Abgrenzung seien folgende Annahmen über die IST-Situation im beispielhaft vorliegenden Unternehmen: Die Aufgaben der betrieblichen Objekte *Fortbildungswesen* und *Personalwesen* werden im Unternehmen bereits durch ein existierendes Lern-Managementsystem (LMS) und ein ERP-basiertes Human-Resource-Managementsystem (HRMS) unterstützt. Die zugrundeliegenden Aufgaben werden in Bezug auf den Flexibilitätsbedarf als relativ stabil angesehen, so dass die genannten Systeme weiter betrieben und lediglich mit der SOA gekoppelt werden sollen. Der *Personalbesetzung* und der *Mitarbeiterqualifizierung* hingegen liegt ein hoher Flexibilitätsbedarf zugrunde. So soll die *Personalbesetzung* zukünftig ggf. eine weitere Prozessausführungsvariante bereitstellen, die auch die Stellenbesetzung durch externe Projektmitarbeiter ermöglicht. Die *Mitarbeiterqualifizierung* soll neben Qualifizierungsmaßnahmen über interne Quellen u. U. zukünftig auch externe Weiterbildungsanbieter in den Prozess einbinden, so dass auch hier ggf. eine weitere Ausführungsvariante erforderlich sein wird. Die beschriebenen Annahmen führen zu der in Bild 7 skizzierten Systemabgrenzung und -kartierung: *Personalbesetzung* und *Mitarbeiterqualifizierung* bilden den *flexiblen Aufgabenraum* und sollen durch eine zu entwickelnde SOA unterstützt werden. Der *herkömmliche Aufgabenraum* umfasst *Fortbildungs-* und *Personalwesen*, welche auf Aufgabenträgerebene durch bestehende Anwendungssysteme gestützt werden sollen. Schnittstellen zwischen den abgegrenzten Aufgabenträumen müssen berücksichtigt werden.
- (3) Eine SOM-AwS-Spezifikation in Form eines VOS und KOS kann unter Verwendung der SOM-Methodik modellgestützt auf Basis der Ergebnisse der Schritte 2 und 3 abgeleitet werden. In Bezug auf die Innensicht einer Aufgabe kann das VOS (Bild 9) der Aktionensteuerung zugeordnet werden, die auf elementaren Aktionen und dem Aufgabenobjekt, repräsentiert durch das KOS, operiert ([10], S. 430).

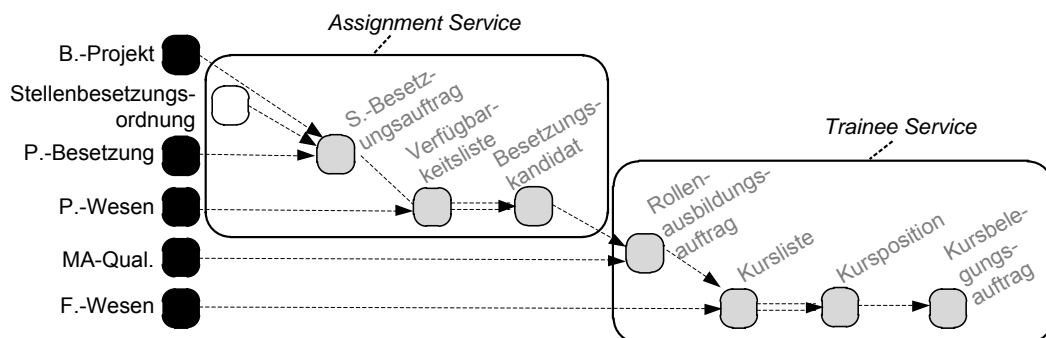




**Bild 9:** Vorgangs-Objektschema der SOA

- (4) Zur Spezifikation der externen Prozessschnittstellen der SOA sind die erforderlichen Schnittstellenbeziehungen zwischen dem *flexiblen* und dem *herkömmlichen Aufgabenraum* zu identifizieren. Sie sind als ein- und ausgehende Nachrichtenflüsse der einzelnen Vorgangsobjekttypen im zuvor abgeleiteten VOS ersichtlich und können direkt in einen elementaren Vorgangs-Service überführt werden. Im vorliegenden Fall ist z. B. der Vorgangsobjekttyp *Verfügbarkeitsanfrage* mit dem ausgehenden Nachrichtenfluss *findCandidates* (Bild 9) Basis des Vorgangs-Services *ermittleVerfügbarkeit* (Bild 11).

Überlappungen der AOs des *flexiblen* und des *herkömmlichen Aufgabenraums* können zum Erfordernis externer Datenschnittstellen führen. Zur Bedarfsanalyse hierfür kann das spezifizierte KOS der SOA herangezogen werden. Weist dieses konzeptuelle Objekttypen auf, deren Attribute auch von den umgebenden AwS verwaltet werden, so muss die Notwendigkeit einer Datenschnittstelle geprüft werden. Diese ist genau dann erforderlich, wenn die Überlappung der AOs auf Typebene mit einer partiellen Identität der AO-Instanzen einhergeht ([10], S. 232f) und Modifikationen der einen AO-Instanz in der anderen AO-Instanz nachgezogen werden müssen. Im Fall des Personal-Wesens könnte ein Beispiel das Ausscheiden eines Mitarbeiters aus dem Unternehmen sein, der zuvor als Besetzungskandidat für ein Projekt (siehe konzeptueller Objekttyp Besetzungskandidat in Bild 10) angeboten worden ist. Soll bei Ausführung einer GP-Instanz auf diese Änderung reagiert werden können, so ist eine Datenschnittstelle zur Abstimmung der AOs erforderlich.

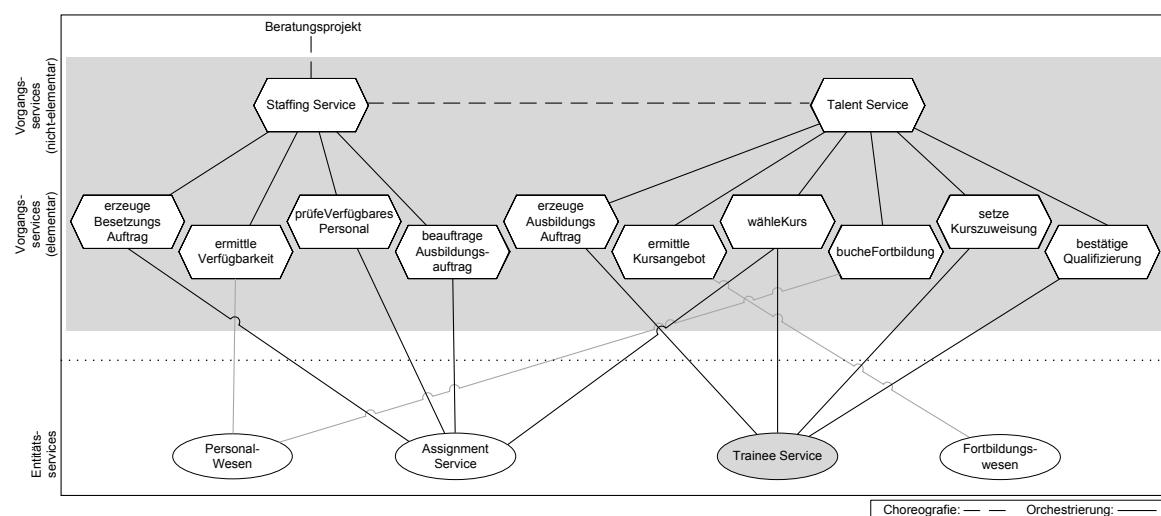


**Bild 10:** Konzeptuelles Objektschema der SOA

- (5) Zugrunde liegt das in [17] skizzierte SOA-Architekturmodell, dessen Bausteine und Beziehungen zur Beschreibung der Innensicht der SOA verwendet werden ([17], S. 294). Hierbei können folgende Ableitungsregeln angewendet werden:

Elementare Vorgangs-Services lassen sich anhand der im VOS spezifizierten Vorgangsobjekttypen identifizieren und abgrenzen. So ist z. B. die Basis der Ableitung für den Vorgangs-Service *erzeugeBesetzungsauftrag* (Bild 11) der zum betrieblichen Objekt *Personal-Besetzung* zugehörige Vorgangsobjekttyp *>Stellenbesetzungsauftrag* (Bild 9).

Nicht-elementare Vorgangs-Services orchestrieren elementare und ggf. weitere nicht elementare Vorgangs-Services zur Realisierung des Vorgangsnetzes eines betrieblichen Objekts ([17], S. 295 f.). Im vorliegenden Beispiel wird jedes der beiden der SOA zugrundeliegenden betrieblichen Objekte durch genau einen nicht-elementaren-Vorgangs-Service gestützt. Diese Art der Abgrenzung ist nicht zwingend erforderlich, vielmehr sind geeignete Zielkriterien für die Abgrenzung das Erreichen einer geeigneten Granularität und aus Außensicht die Abbildung eines zusammenhängenden und sinnvoll gruppierten Aufgabenraums.



**Bild 11: Service-orientierte Architektur**

Grundlage der Ableitung von Entitäts-Services ist das zuvor im KOS spezifizierte AO. Zur Ableitung und im Hinblick auf eine angemessene Granularität werden einzelne fachlich zusammenhängende konzeptuelle Objekttypen zu einem gemeinsamen Entitäts-Service zusammengefasst ([17], S.307 f.). Aus Aufgabenmodell-Sicht führt eine Zerlegung des Gesamt-AOs zu partiell überlappenden Teil-AOs. Aufgrund der Entkopplung von Entitäts-Services und darauf operierenden Vorgangs-Services führt dies auf Ebene der SOA nicht zwangsläufig zu dem Erfordernis einer redundanten Datenhaltung, sondern kann, wie im Beispiel des Entitäts-Service Assignment Service (Bild 11) gezeigt, auch durch einen Zugriff von mehreren Vorgangs-Services auf einen Entitäts-Services realisiert werden. Vorgangsnetze, die von unterschiedlichen betrieblichen Objekten erbracht werden, sind grundsätzlich autonom, so dass ihre Schnittstellen per Choreografie realisiert werden.

## 4 Stand der Literatur

Ansätze zum Entwurf von SOA in der Literatur lassen sich grob in zwei Kategorien unterteilen: Pattern und Prinzipienbasierte Ansätze (z. B. [8], [9] oder [3]) umfassen allgemeine Designprinzipien für SOA, deren Anwendung und Kombination dem Entwickler

überlassen bleibt ([19], S. 421). Hierarchisch strukturierte Ansätze zum Service-Design hingegen beschreiben die erforderlichen Schritte zum fachlichen Entwurf von SOA auf mehreren aufeinander aufbauenden Ebenen ([19], S. 421). Der vorliegende Ansatz zum Entwurf partieller SOA ist der letztgenannten Kategorie hierarchischer Ansätze zuzuordnen. Diese lassen sich grob weiter untergliedern in *Top-Down*-, *Bottom-Up* und *Meet-in-the-middle*-Entwicklungsansätze ([18], S. 631 f.). Während *Top-Down*-Ansätze bei der Ableitung von Services aus fachlicher Sicht vorgehen, ist der Ausgangspunkt von *Bottom-up*-Ansätzen die bestehende Anwendungssystem- und Service-Landschaft im Unternehmen. *Meet-in-the-Middle*-Ansätze hingegen berücksichtigen beim Entwurf von SOA sowohl eine fachliche Sicht als auch eine Sicht auf die bestehende AwS-Landschaft im Unternehmen. Dem vorliegenden Ansatz liegt im Kern eine *Top-Down*-getriebene Entwurfsrichtung zur Ableitung partieller SOA zugrunde. Da die GP-Modellebene eines Unternehmens hierbei als Ausgangspunkt für den Ansatz verwendet wird, leistet er einen Beitrag zum Ziel des IT / Business-Alignment. Die ergänzende Betrachtung der bestehenden IT-Landschaft (*Bottom-Up*) durch den Ansatz begründet sich aus der Tatsache, dass eine SOA in der Regel nicht auf der grünen Wiese entsteht, sondern sich als partielles Glied in die bestehende Anwendungssystemlandschaft eines Unternehmens einfügt. Durch die um eine *Bottom-Up*-Sicht ergänzte *Top-Down*-getriebene Entwurfsstrategie lässt sich der vorliegende Ansatz somit der Kategorie *Meet-in-the-Middle* zuordnen.

Zur Klassifikation und Untersuchung von Eigenschaften von SOA-Entwicklungsansätzen existieren zahlreiche Literaturbeiträge (z. B. [4], [5], [12], [15], [21]). Auf dieser Basis und auf Grundlage einer eigenen umfassenden Literaturanalyse lässt sich feststellen, dass eine Vielzahl bereits bestehender SOA-Entwicklungsansätze in die Kategorie *Meet-in-the-middle* fallen. Nur wenige gehen allerdings ins Detail ([12], S. 40) und beschreiben eine umfassende modellbasierte Methodik zur Ableitung von SOA, wie es z. B. bei dem von Arsanjani et al. veröffentlichten Ansatz *Service-oriented modeling and architecture* (SOMA) der Fall ist [1]. Genau wie der vorliegende Ansatz sieht SOMA zwar auch ein Mapping zwischen der GP-Modell- und der AwS-Ebene vor ([1], S. 388). Betrachtet werden hierbei allerdings ausschließlich die Prozessschnittstellen zwischen den beteiligten AwS bzw. Services. Überlappende Aufgabenobjekte, die ggf. zum Erfordernis gesonderter Datenschnittstellen führen (siehe Fallbeispiele in Kapitel 3 und [17], Kapitel 14.4), werden nicht berücksichtigt. Hier liegt ein Vorteil des vorliegenden Ansatzes, da gerade die Daten, auf denen betriebliche Aufgaben operieren, zentraler Betrachtungsgegenstand des Ansatzes sind.

Viele Ansätze zum Entwurf von SOA folgen dem Konzept der *Model Driven Architecture* (MDA) [14]. Dieses Konzept unterscheidet die Modellebenen CIM (*Computation Independent Model*), PIM (*Platform Independent Model*) und PSM (*Platform Specific Model*). Der vorliegende Ansatz beruht zwar nicht auf diesem Konzept, kann aber auf verschiedenen Ebenen mit diesem verglichen werden. So ergibt sich eine Differenzierung des Ansatzes gegenüber MDA-basierten Ansätzen zur Entwicklung von SOA insbesondere auf Ebene der Modelltransformation von CIM zu PIM. Die dem vorliegenden Ansatz zugrundeliegende SOM-Methodik dient der ganzheitlichen Modellierung betrieblicher Systeme, wobei die initiale fachliche AwS-Spezifikation (MDA-Ebene PIM) zur Automatisierung der modellierten Geschäftsprozesse direkt aus dem GP-Modell (MDA-Ebene CIM) abgeleitet werden kann. Nur wenige MDA-Ansätze (z. B. [7]) betrachten und unterstützen diese Ebene im Detail. I. d. R. liegt der Schwerpunkt der Betrachtung eher auf der Modelltransformation zwischen

den Ebenen PIM und PSM, z. B. indem ausgehend von BPMN-Workflowdiagrammen (MDA-Ebene PIM) die Generierung von auf einer Process-Engine ausführbaren BPEL-Prozessen unterstützt wird (z. B. [11]).

## 5 Diskussion

Der Beitrag zeigt, wie anhand des Aufgabenmodells in wenigen Schritten ein umfassendes Verständnis von SOA entwickelt werden kann, welches sowohl das Lösungsverfahren als auch das bei SOA oft vernachlässigte Aufgabenobjekt mit einbezieht. Anschließend wird der Ansatz anhand einer Fallstudie illustriert und am Stand der Literatur gespiegelt.

Der Modellierungsansatz führt zu einer partiellen SOA und ist dort von Vorteil, wo die anderen mit SOA verfolgten Ziele als nachrangig betrachtet werden können. Gleichwohl dürfte dieser Fall häufig vorkommen, sei es alleine aus Gründen der Wirtschaftlichkeit ([22], S. 53ff.). Hier treten die Schnittstellen zu den herkömmlichen Systemen in den Vordergrund, denen sich der vorliegende Ansatz besonders annimmt.

Derzeit wird an der Werkzeugunterstützung ([6], S. 367ff.) gearbeitet, eine Bedingung, um untersuchen zu können, ob und wie der Ansatz skaliert. Eine größere Fallstudie wurde dazu in [17] beschrieben. Schließlich ist die Annahme der Vollautomatisierung aufzugeben und zu untersuchen, welche Einflüsse sich durch Einbeziehung des Menschen als weiteren Aufgabenträger ergeben. Zudem ist zu untersuchen, welche weiteren Architekturkonzepte neben SOA zur Deckung von Flexibilitätsbedarfen in Frage kommen und in welcher Beziehung diese zu einer partiellen SOA stehen.

## 6 Literatur

- [1] Arsanjani, A; Ghosh, S; Allam, A; Abdollah, T; Gariapathy, S; Holley, K (2008): SOMA: a method for developing service-oriented solutions. *IBM Systems Journal* 47(3):377-396.
- [2] Bartmann, D; Bodendorf, F; Ferstl, OK, Sinz, EJ (2011): Merkmale, Systemarchitekturen und Management hochflexibler Geschäftsprozesse. In: Sinz, EJ; Bartmann, D; Bodendorf, F; Ferstl, OK (Hrsg.), *Dienstorientierte IT-Systeme für hochflexible Geschäftsprozesse*. University of Bamberg Press, Bamberg.
- [3] Bell, M (2010): *SOA Modeling Patterns for Service-Oriented Discovery and Analysis*. Wiley & Sons, Hoboken, NJ.
- [4] Birkmeier, D; Klöckner, S; Overhage S (2008): Zur systematischen Identifikation von Services: Kriterien, aktuelle Ansätze und Klassifikation. In: Loos, P (Hrsg.), *Modellierung betrieblicher Informationssysteme (MobIS 2008). Modellierung zwischen SOA und Compliance Management*. Saarbrücken.
- [5] Börner, R; Goeken, M (2009): Identification of Business Services Literature Review and Lessons Learned. In: *Proceedings of the 15th Americas Conference on Information Systems 2009*. San Francisco, CA, USA.
- [6] Bork, D; Sinz, EJ (2011): Ein Multi-View-Modellierungswerkzeug für SOM-Geschäftsprozessmodelle auf der Basis der Meta-Modellierungsplattform ADOxx. In: Sinz, EJ; Bartmann, D; Bodendorf, F; Ferstl OK (Hrsg.), *Dienstorientierte IT-Systeme für hochflexible Geschäftsprozesse*. University of Bamberg Press, Bamberg.

- [7] De Castro, V; Marcos, E; Vara, JM (2011): Applying CIM-to-PIM model transformations for the service-oriented development of information systems. *Information and Software Technology* 53(1):87-105.
- [8] Erl, T (2008): SOA. Principles of Service Design. Prentice Hall PTR, Upper Saddle River, NJ.
- [9] Erl, T (2009): SOA. Design Patterns. Prentice Hall PTR, Upper Saddle River, NJ.
- [10] Ferstl, OK; Sinz, EJ (2008): Grundlagen der Wirtschaftsinformatik. Oldenbourg, München.
- [11] Giner, P; Torres, V; Pelechano, V (2007): Bridging the Gap between BPMN and WS-BPEL. M2M Transformations in Practice. In: *Proceedings of the 3rd International Workshop on Model-Driven Web Engineering (MDWE 2007)*. Como, Italien.
- [12] Gu, Q; Lago, P (2010): Service Identification Methods: A Systematic Literature Review. In: Hutchison, D et al. (Hrsg.), *Towards a Service-Based Internet - Third European Conference, ServiceWave 2010*. Ghent, Belgium.
- [13] Heinrich, LJ (1993): Wirtschaftsinformatik. Einführung und Grundlegung. Oldenbourg, München.
- [14] Miller, J; Mukerji, J (2003): MDA Guide Version 1.0.1. <http://www.omg.org/cgi-bin/doc?omg/03-06-01>. Abgerufen am: 18.09.2011.
- [15] Kohlborn, T; Korthaus, A; Chan, T; Rosemann, M (2009): Identification and Analysis of Business and Software Services - A Consolidated Approach. *IEEE Transactions on Services Computing* 2(1):50-64.
- [16] Kosiol, E (1976): Organisation der Unternehmung. Betriebswirtschaftlicher Verlag Gabler, Wiesbaden.
- [17] Krücke, A; Sinz, EJ (2011): Entwurf partieller SOA auf der Grundlage von Geschäftsprozessmodellen. In: Sinz, EJ; Bartmann, D; Bodendorf, F; Ferstl OK (Hrsg.), *Dienstorientierte IT-Systeme für hochflexible Geschäftsprozesse*. University of Bamberg Press, Bamberg.
- [18] Papazoglou, MP (2008): Web Services. Principles and Technology. Pearson, Boston.
- [19] Patig, S; Wesenberg, H (2009): Role of Process Modeling in Software Service Design. In: Hutchison, D et al. (Hrsg.), *Proceedings of the 7th International Joint Conference on Service-Oriented Computing (ICSOC-ServiceWave '09)*. Stockholm, Schweden.
- [20] Peltz, C (2003): Web Services Orchestration and Choreography. In: *IEEE Computer* 36(10): 46-52.
- [21] Ramollari, E; Dranidis, D; Simons, AJH (2007): A Survey of Service Oriented Development Methodologies. In: Gorton, S; Solanki, M; Reiff-Marganiec, S (Hrsg.), *Proc. of 2nd Young Researchers' Workshop on Service Oriented Computing*. Leicester, U.K.
- [22] Wagner, D; Leunig, B; Suchan, C; Frank, J; Ferstl, OK (2011): Klassifikation von Geschäftsprozessen anhand ihres Flexibilitätsbedarfs. In: Sinz, EJ; Bartmann, D; Bodendorf, F; Ferstl, OK (Hrsg.), *Dienstorientierte IT-Systeme für hochflexible Geschäftsprozesse*. University of Bamberg Press, Bamberg.